

x86Launcher

What is it?

x86Launcher is graphical game browser and launcher for the IBM/PC compatibles running MS/PC/DR-DOS or equivalent.

It runs on real DOS hardware **or** from within Dosbox, either emulated or real physical system; it is **not** a Window/Mac/Linux application like [launchbox](#) or [RetroArch](#).

x86Launcher is a 16bit, real-mode application and will work on any hardware compatible with the following:

- Intel 8086 (or compatible) or higher x86 processor
- 640K of memory (no XMS or EMS required)
- VGA card with 256KB memory and VESA 1.1/1.2 support (must support 640x400 @ 256 colours)

x86Launcher is the DOS equivalent of my other projects:

- [PC98Launcher](#) - A game browser and launcher for the NEC PC9801 series of computers
- [X68KLauncher](#) - A game browser and launcher for the Sharp X68000 series of computers

What does it do?

- It creates a browseable list of all the game directories on your drive(s)
- It can load and display additional metadata per game (i.e developer, genre, release date etc)
- If metadata is available, it can also do:
 - Search/Filter games by publisher or developer
 - Search/Filter by genre (e.g. FPS, Fighting, Adventure, Strategy)
 - Search/Filter by series (e.g. Ultima 1-8, the DOOM series, Advanced Dungeons and Dragons RPG's)
 - Load and display screenshots or artwork per game (i.e box art, screenshots, etc)
- It can export an audit file of all the found games
- It can launch any game for which a start file is either found (e.g. start.bat), or which has been defined in metadata (e.g. run.com, go.exe, etc)

Sample Use

The image below takes you to a sample movie of the launcher application in action via **YouTube**:

TBD

Click to the video above to see the launcher in action

What does it look like?

Starting up and scanning for games:

TBD

Main user interface:

TBD

Browsing game artwork:

TBD

TBD

How do I run it?

You will need a system that meets these requirements:

- An IBM/PC or compatible with 8086 or higher x86 processor (clones processors and compatible - such as NEC V20/V30 *should* work)
- A minimum of 640K of RAM
- A hard drive (technically it can run from floppy, but it wouldn't be practical!)
- A VGA graphics card that can support VESA 640x400 in 256 colours (most cards can do this either natively, or with an add-on like UniVBE that adds VESA support to those cards without it). 640x400 resolution means that you only need 256K of VGA memory.

You then need to copy the zipped binaries (see below) to a location on your PC-98 hard drive and unzip them to a directory of your choice.

The directory should have the following after unzipping:

- launcher.exe
- launcher.ini
- l.bat
- assets\font8x16.bmp
- assets\logo.bmp
- assets\light*.bmp

You don't need to set anything in config.sys or autoexec.bat.

Edit launcher.ini with a normal text editor and set the **gamedirs** variable to all of the locations of games on your hard drive(s).

The following settings are available for launcher.ini:

- gamedirs=C:\Path1,C:\Path2,D:\Path3 - List the directories which contain your game subdirectories
- verbose=0|1 - Enable text mode logging for troubleshooting purposes
- savedirs=0|1 - Save the scraped list of games to a text file at start
- preload_names=0|1 - For each found game, attempt to load the metadata file to get its real name. This will slow initial scraping down.
- keyboard_test=0|1 - Before starting the UI, prompt the user to do a quick input test

If you have your games under folders such as **C:\Games\Arkanoid** and **C:\Games\Dark** for example, then you only need to add the path **C:\Games**. You may add up to 16 comma seperated game paths, and these can be for different drives if you wish.

Just run **l.bat** to start the application.

Binaries

The latest pre-compiled binaries as well as pre-made metadata and screenshots can be found on my website under the [Tech Stuff](#) wiki:

- www.target-earth.net - Tech Stuff wiki

How to build

The code for x86Launcher is all standard ANSI C, built using Open Watcom 2.

A full guide of how to set up a build environment is on my wiki:

- www.target-earth.net - IBM/PC DOS Dev tools wiki

Adding metadata & images

Metadata files

You can add additional metadata to your game by placing a text file in each game directory named **launch.dat**.

The contents of this file is as follows:

```
[default]
name=
developer=
midi_mpu=
midi_serial=
year=
genre=
images=
series=
start=
alt_start=
```

The fields **name**, **developer** and **genre** are all simple text, maximum of 32 characters.

The fields **midi_mpu** and **midi_serial** are integers, where **1** means supported and any other value means unsupported.

The **images** field is a comma separated list of the artwork available. The path is relative to the game directory, so **image01.bmp** will be expanded to include the drive and path that the game is located under (C:\Games\gamename\image01.bmp, for example).

Images will be shown in the order they are listed, so place the image you want shown by default as the first item in the list.

The **series** field is a text name of the larger game series in which the game is based, useful for those games in which there are more than one game (Doom and Doom II, for example). You can use the **filter** option within the application to find all games within the same series, as long as they are tagged up with the correct metadata.

The **genre** field notes the type of gameplay within the game (RPG, Action, Sports, Puzzle, etc). As with **series** you can use the application **filter** facility to restrict the display of games to just one type of genre if desired. Again, you need to set this metadata in order to make use of it within the interface, otherwise it will be treated as blank.

Finally, the **start** and **alt_start** data determines which exe should be launched when the game is selected. By convention I would recommend the main game executable or batch file is set for the **start** value, and any stand-alone cinematics, or sound configuration utility is set for the **alt_start** value.

An example is shown below:

```
[default]
name=Duke Nukem 3D
developer=3D Realms
publisher=GT Interactive
midi_mpu=1
midi_serial=1
year=1996
genre=FPS
images=cover.bmp,screen1.bmp,screen2.bmp,box.bmp
series=Duke Nukem
start=DUKE3D.EXE
alt_start=SETUP.EXE
```

You could then browse to "Duke Nukem 3D" in the game list, search for it by games published by "GT Interactive", developed by "3D Realms", or all titles in the series "Duke Nukem".

Supplementary Metadata

The metadata file can have several other section added to it, which are entirely optional, these sections are:

[sound], [video] and [cpu].

Sound Metadata

The sound metadata block can have zero, one or more of the following entries:

```
[sound]
beeper=
adlib=
soundblaster=
mt32=
gm=
tandy=
ultrasound=
disney=
covox=
```

All entries are set to either 0 or 1 to indicate support for that particular audio device.

Sound Metadata

The video metadata block can have zero, one or more of the following entries:

```
[video]
text=
cga=
ega=
vga=
tandy=
hercules=
svga=
```

All entries are set to either 0 or 1 to indicate support for that particular video display device.

CPU Metadata

The CPU metadata block can have zero, one or more of the following entries:

```
[cpu]
xms=
ems=
```

dpmi=
8086=
286=
386=
486=
586=

All entries are set to either 0 or 1.

For **EMS** and **XMS** a setting of 1 indicates that the title requires DOS memory of that particular type.

For **DPMI** a setting of 1 indicates that the title is a 32bit protected mode game *requiring* a 386 or above in order to run.

For the various x86 cpu types, a setting of 1 indicates that the title will *run adequately* on a processor of that type.

Converting images to useable BMP files

The BMP files used by the application need to be 8bpp but of a limited palette so that the user interface does not suffer from colour issues (approximately 40 colours are reserved by the user interface and fonts).

You can use any image processing application you want, but it must output images of the following specifications:

- BMP
- Uncompressed
- 8bpp, indexed/paletted colour
- Maximum of 208 colours
- No larger than 320x200 (but they may be smaller in either dimension, if desirable, e.g. for vertical boxart)

If you have the [ImageMagick](#) tools available on your system, you can batch convert files using the following syntax:

```
convert INPUT.JPG -resize 320x200 -depth 8 -colors 208 -alpha OFF -compress none BMP3:OUTPUT.BMP
```

Note: *If you do not reduce the number of active colours in use in your screenshots and box art, the images will still show, but they may display incorrectly.*

Automating Image Conversion

A script to automate the image conversion (for Linux systems) is included in the **scripts/** directory.